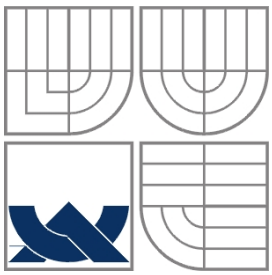


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

VYTVOŘENÍ VYUKOVÝCH PODKLADŮ PRO PRÁCI VE VÝVOJOVÉM PROSTŘEDÍ LOGO! SOFT COMFORT

TEACHING DOCUMENTS CREATIONS FOR WORK IN THE DEVELOPMENT SYSTEM LOGO! SOFT COMFORT

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

AUTOR PRÁCE **STANISLAV KOLÁŘ**
AUTHOR

VEDOUCÍ PRÁCE **ING. TOMÁŠ MARADA, PH.D.**
SUPERVISOR

BRNO 2007

ZADÁNÍ ZÁVĚREČNÉ PRÁCE

(na místo tohoto listu vložte originál a nebo kopii zadání Vaš práce)

LICENČNÍ SMLOUVA

(na místo tohoto listu vložte vyplněný a podepsaný list formuláře licenčního ujednání)

ABSTRAKT

Mým úkolem je seznámit se s logickým programováním v softwaru LOGO! Soft Comfort, popsat ho a navrhnout ukázkové úlohy. V této práci bych chtěl studentům pomoci pochopit tuto problematiku logického programování. Chtěl bych aby díky mé práci bylo pro budoucí studenty programování tohoto PLC jednodušší a snazší. Představím a popíšu zde výše zmiňovaný software, jeho prvky a vše budu demonstrovat na navržených příkladech.

ABSTRACT

My task is to become acquianted with logical progamming in software LOGO!Soft Comfort, describe it and project demonstration excercises. In this schoolwork I would like to assist in understanding of questions about logical progamming. My idea is that my work will help to the future students to make progamming of this PLC much more simple and easy. I will present and describe this software and its parts and everything I will show on the proposed examples.

KLÍČOVÁ SLOVA

PLC, LOGO!.

KEYWORDS

PLC, LOGO!.

Obsah:

1 ÚVOD

Úkolem je vytvořit výukové podklady pro vývojový software LOGO! Soft Comfort, představit zde tento logický modul, jeho vývojové prostředí a předvést nějaké ukázkové úlohy v praxi. Tato práce vzniká z důvodů zakoupení těchto nových modulů do výukových učeben programovatelných automatů.

Obr. 1 Logický modul LOGO!

LOGO! - je to logický modul, který se využije pro řešení u jednodužších aplikací, kde se nachází například časovče, čítače, pulzní relé apod. Využijeme to hlavně, když se takových prvků vyskytuje u nějaké aplikace více. Může ovládat např. kompresory, výtahy, závory, apod. Naprogramovat tento modul můžeme přímo na displeji nebo můžeme použít software k tomuto určený.

Tento logický modul má mnoho logických a časových funkcí, má také mnoho speciálních funkcí pro pokročilejší využití. Může mít jak vstupy analogové tak i digitální. Je možné LOGO! rozšířit o jiné moduly s dalšími vstupy a výstupy.

Obr. 2 LOGO! s přídavnými moduly

Popisovat se bude konkrétně model LOGO! 0BA5. Tato verze je zatím nejnovější a má mnoho vylepšení oproti předchozím verzím. Obsahuje 37 funkcí, může se zapojit až 130 bloků, není zde omezen počet časovačů, čítačů a pod. tudíž může být program bez omezení, dokáže uložit aktuální hodnoty v případě výpadku napájení a mnoho dalšího dokáže tato nová verze.

Obr. 3 LOGO! S přídavnými moduly se vstupy a výstupy

LOGO! Je k dispozici v následujících verzích:

Symbol	Označení	Napájecí napětí	Vstupy	Výstupy
	LOGO! 12/24 RC	12/24 V DC	8 digitálnich (1)	4 reléové (10 A)
	LOGO! 24	24 V DC	8 digitálnich (1)	4 polovodičové 24 V / 0,3 A
	LOGO! 24RC (3)	24 V AC 24 V DC	8 digitálnich	4 reléové (10 A)
	LOGO! 230RC (2)	115…240 V AC/DC	8 digitálnich	4 reléové (10 A)
	LOGO! 12/24RCo	12/24 V DC	8 digitálnich (1)	4 reléové (10 A)
	LOGO! 24o	24 V DC	8 digitálnich (1)	4 polovodičové 24 V / 0,3 A
	LOGO! 24RCo(3)	24 V AC 24 V DC	8 digitálnich	4 reléové (10 A)
	LOGO! 230RCo (2)	115…240 V AC/DC	8 digitálnich	4 reléové (10 A)

Obr. 4 Tabulka vychází z manuálu LOGO! 0BA5

(1): Alternativně je možné použít: 2 analogové vstupy (0 ... 10V) a 2 rychlé vstupy.
(2): Verze 230 V AC: Dvě skupiny z nichž každá má 4 vstupy. Každý vstup v jedné skupině musí být připojen ke stejné fázi. Je možné propojit skupiny s různou fází.
(3): Digitální vstupy je možné provozovat a činností P nebo N.

K LOGO! můžeme připojit následující rozšiřovací moduly:

Symbol	Název	Napájení	Vstupy	Výstupy
	LOGO! DM 8 12/24R	12/24 V DC	4 digitální	4 reléové (5A)
	LOGO! DM 8 24	24 V DC	4 digitální	4 polovodičové 24 V / 0,3 A
	LOGO! DM 8 24R (3)	24 V AC/DC	4 digitální	4 reléové (5A)
	LOGO! DM 8 230R	115…240 V AC/DC	8 digitálnich	4 reléové (5A)
	LOGO! DM 16 24	24 V DC	8 digitálnich	8 polovodičových 24 V / 0,3 A
	LOGO! DM 16 24R	24 V DC	8 digitálnich (1)	8 reléových (5A)
	LOGO! DM 16 230R	115…240 V AC/DC	8 digitálnich (4)	8 reléových (5A)
	LOGO! AM 2	12/24 V DC	2 analogové 0 až 10 V nebo 0 až 20 mA (2)	nemá
	LOGO! AM 2 PT100	12/24 V DC	2 Pt100 -50 °C až +200 °C	nemá
	LOGO! AM 2 AQ	24 V DC	nemá	2 analogové 0 až 10 V DC

Obr. 5 Tabulka vychází z manuálu LOGO! 0BA5

(1): V rámci vstupu nejsou povoleny různé fáze.
(2): Volitelně je možné připojit 0 ... 10 V, 0 ... 20 mA.
(3): Digitální vstupy je možné provozovat s činností P nebo N.
(4): Dvě skupiny z nichž každá má 4 vstupy. Každý vstup v jedné skupině musí být připojen ke stejné fázi. Je možné propojit skupiny s různou fází.

K LOGO! můžeme připojit následující komunikační moduly:

Symbol	Název	Napájení	Vstupy
	LOGO! CM AS Interface	24 V DC	čtyři vstupy následujici po fyzických vstupech LOGO! (In ... In+3)
	LOGO! CM EIB/KNX	24 V AC/DC	max. 16 virtuálnich digitálnich vstupů (I); max. 8 virtuálnich analogových vstupů (AI)

Obr. 6 Tabulka vychází z manuálu LOGO! 0BA5

2 UŽIVATELSKÉ PROSTŘEDÍ

Abychom mohly pracovat s tímto programem LOGO! Soft Comfort, předpokládá se znalost operačního systému Windows a vytváření funkčních blokových diagramů. Pro přenášení programu z počítače do modulu LOGO! je potřeba propojovací kabel.

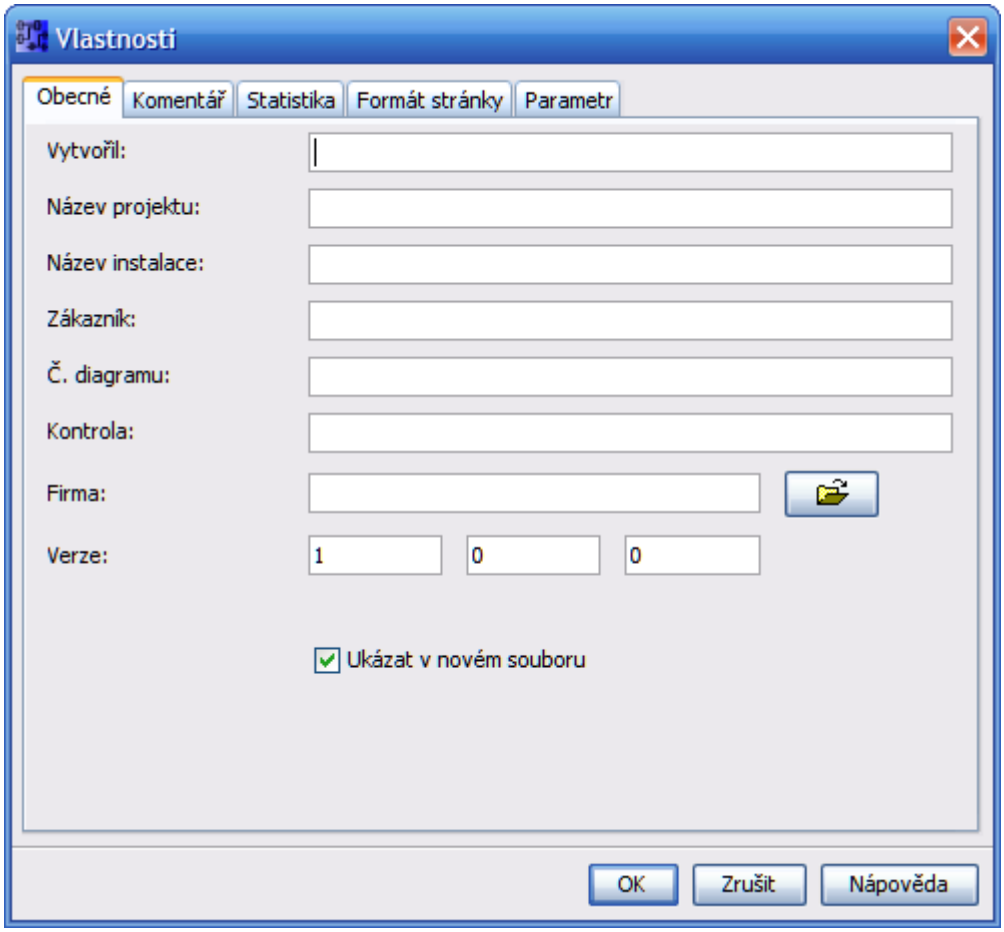
Tato kapitola vychází z nápovědy LOGO! Soft Comfort.

2.1 Uživatelské rozhraní

Po spuštění programu se otevře prázdné uživatelské rozhraní a musíme si otevřít nový projekt za pomocí této ikony:

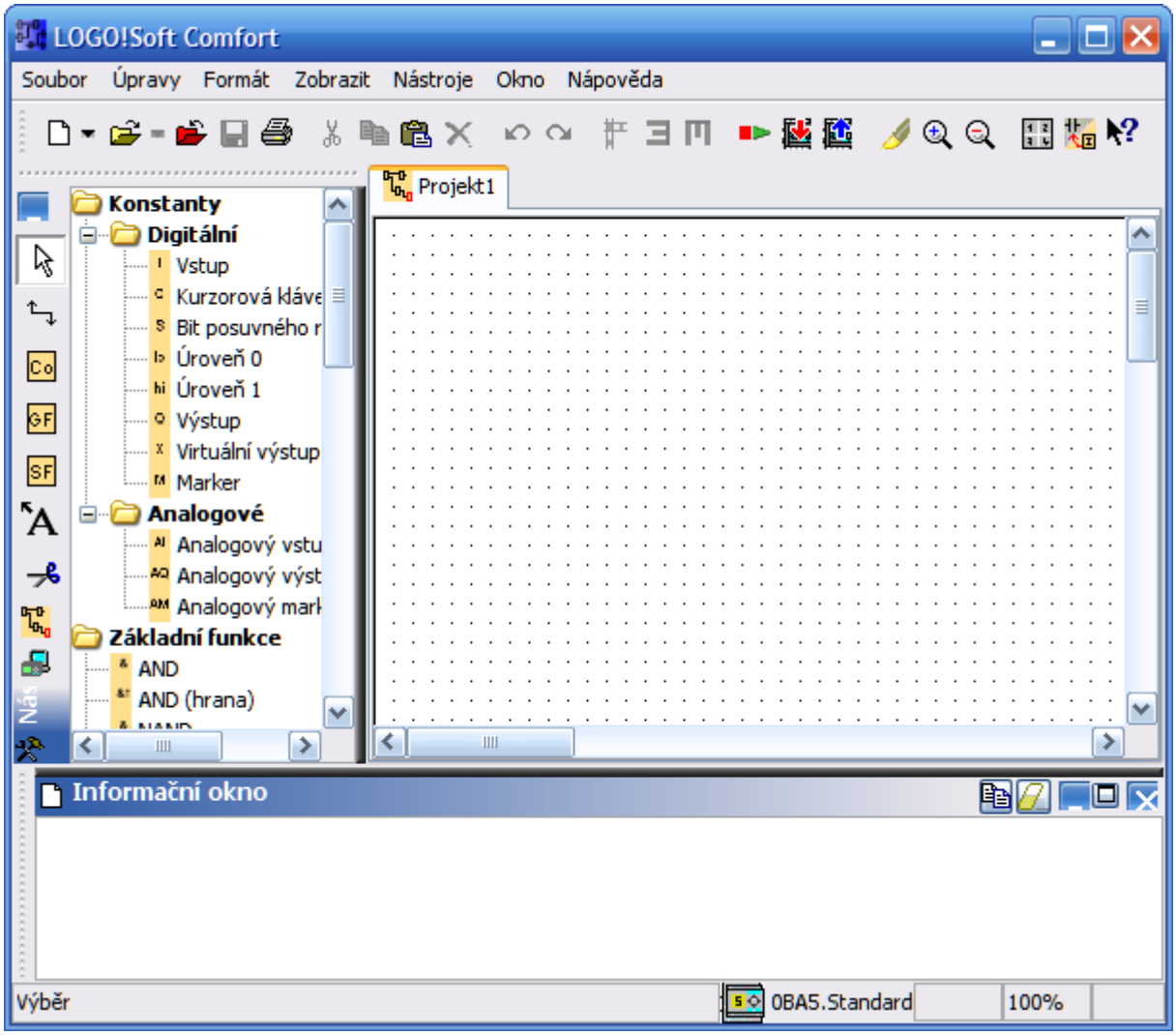
Obr. 7 Ikona pro otevření nového projektu

Po kliknutí na tuto ikonu se otevře okno vlastnosti, kde si můžeme zadat potřebné parametry:



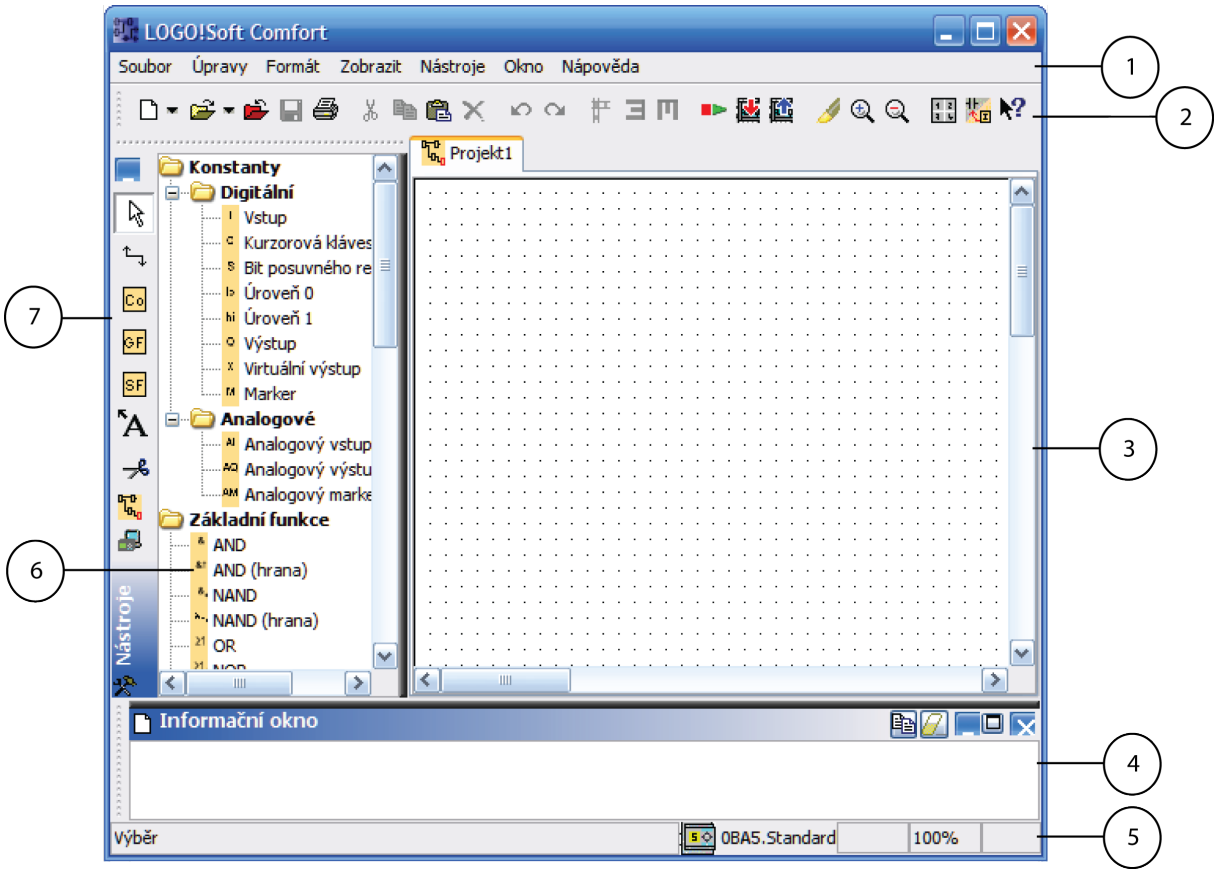
Obr. 8 Okno Vlastnosti

Po zádání příslušných parametrů zmáčkneme tlačítko OK a po té se dostaneme do programového prostředí kde se vytváří samotné projekty.



Obr. 9 Okno celého programu

Na programovací ploše jsou po pravé a spodní straně posouvací lišty pro snadnější přehled velkých obvodů.

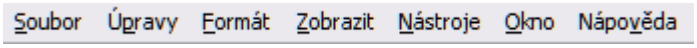


Obr. 10 Vlastní okno programu

1. Lišta menu
2. Standartní nastrojová lišta
3. Programovací rozhraní
4. Informační okno
5. Stavový řádek
6. Konstanty a konektory, základní funkce a speciální funkce
7. Panel nástrojů programování

2.1.1 Lišta menu

Nachází se v horní části okna LOGO! Soft Comfort a nachází se zde různé příkazy pro editaci a správu programu. Jsou zde také funkce pro definování implicitních nastavení a pro přenos programů.



Obr. 11 Lišta menu

2.1.2 Standartní panel nástrojů

Naleznem ji nad programovací plochou. Když spustíme LOGO! Soft Comfort tak se nám zobrazí pouze zjednodušený panel nástrojů kde najdeme pouze základní funkce.

Standartní panel nástrojů poskytuje přímý přístup k základním funkcím LOGO! Soft Comfort. Po otevření nového projektu se nám zobrazí kompletní standartní panel nástrojů.

Obr. 12 Standartní panel nástrojů

Tyto ikony slouží k tvorbě nového programu nebo k načtení, uložení a vytištění stávajícího programu, k vyjmutí/kopírování a vložení objektů, nebo ke spuštění přenosu dat do jednotky LOGO! a naopak.

2.1.3 Programovací rozhraní

V programovacím rozhraní tvoříme projekt samotný. Zde vkládáme potřebné konstanty, základní funkce a speciální funkce z kterých nám vzniká schéma zapojení. Schéma může být uspořádano buď jako kontaktní schéma – LAD a nebo jako funkční bloky – FBD. Zde budeme používat funkční bloky – FBD.

2.1.4 Informační okno

Je umístěné ve spodní části programovací plochy. Zobrazuje informace a poznámky a také typy modulů LOGO! doporučené pro vytvořený program.

2.1.5 Stavový řádek

Nacházíé se ve spodní části programovacího okna. V tomto stavovém okně je zobrazen sktivní nástroj, status programu, nastavený faktor lupy, číslo strany schématu a zvolené zařízení LOGO!.

Obr. 13 Stavový řádek

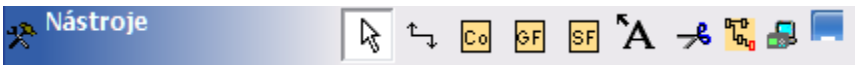
1. Informační pole. Zobrazuje aktuálně použitý nástroj.
2. Zobrazuje vybraný typ modulu LOGO!, pokud žádný typ není vybraný, stačí na to dvakrát kliknout a otevře se okno, kde si můžete vybrat typ pro který modul ten program děláte.
3. Zobrazuje aktuální stav programu. Toto pole je prázdné, pokud jste neprovedli ve schématu žádné změny oproti naposledy uloženému schématu. Text „mod“ označuje, že program byl od posledního uložení modifikován.
4. Zobrazuje aktuální nastavený faktor lupy.
5. Toto poslední pole zobrazuje aktuální stránku schématu.

2.1.6 Konstanty a konektory, základní funkce a speciální funkce

Tady se nalézají všechny dostupné funkční bloky, které můžeme použít při tvorbě projektu.

2.1.7 Panel nástrojů programování

Tento panel se nalézá v levé části programovacího okna. Jeho ikony slouží k přepínání do jiných editovacích režimů nebo pro rychlou a snadnou tvorbu nebo editaci programu.



Obr. 14 Panel nástrojů programování

Panel nástrojů můžeme pomocí myši přesunout na jiné místo. Panel nástrojů se vždy po zavření zasune na horní stranu lišty menu.

2.1.8 Panel nástrojů simulace a stavové okno

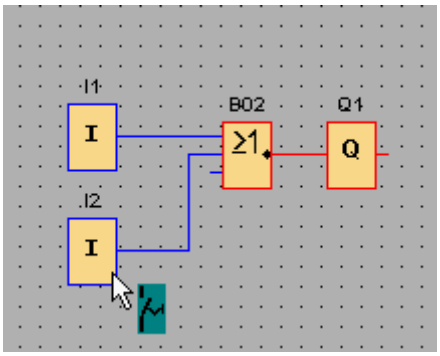
Simulace se spouští jednoduše klávesou F3 a nebo v liště menu příkazem: Nástroje → Simulace. Po spuštění simulačního režimu se objeví panel nástrojů. Na tomto panelu najdeme:

- Ikony (např. spínače) pro operátorskou kontrolu vstupů.
- Ikonu pro simulaci výpadku napájení, pro testování spínací odezvy s odkazem na charakteristiky remanence po výpadku napájení.
- Ikony (např. žárovky) pro monitorování výstupů.
- Ikony řízení simulace
- Ikony časové kontroly

Obr. 15 Panel simulace

Abychom viděly stav signálů ve stavovém okně je zapotřebí mít aktivovanou tuto možnost. Provádí se to pomocí příkazu: Nástroje → Možnosti: Simulace.

Barva čáry nám udává hodnotu signálu. Implicitně je nastavena červená barva na hodnotu 1 a modrá barva pro hodnotu 0.



Obr. 16 Příklad barvy signálu

2.2 Tvorba programu

Při tvorbě programu nám postačí pouze panel nástrojů programování a standartní panel nástrojů. Do programovacího rozhraní budeme vkládat funkční bloky potřebné pro konkrétní úlohu a jednou za čas je potřeba si to uložit.

2.2.1 Tvorba nového programu

Po spuštění LOGO! Soft Comfort můžeme začít s tvorbou nového projektu. Klikneme na ikonu Nový na standartní nástrojové liště (Obr. 7) a po možném vyplnění tabulky vlastností (Obr. 8) můžeme tvořit nový projekt.

Můžeme si také vybrat v jakém editoru budeme tvořit projekt. Jestli FBD nebo LAD. Vyberem tak, že klikneme na malou šipku na pravé straně od ikony Nový a otevřeme tak LAD nebo

FBD editor. Dá se také nastavit implicitní editor pomocí příkazu: Nástroje → Možnosti: Standartní editor. Budeme zde využívat pouze FBD editoru.

2.2.2 Výběr bloků

Při tvorbě schématu obvodu je prvním krokem výběr bloků pro konkrétní obvod. Musíme si uvědomit co chceme vytvořit a zamyslet se co k tomu budeme potřebovat. Pak budeme vkládat vstupy/výstupy a standartní/speciální funkční bloky.

Výběr bloků provádíme kliknutím na tlačítko Konstanty/Konektory na programovacím panelu nástrojů. To otevře nabídku konstant a konektorů ve spodní části programového rozhraní. Je zde také tlačítko Základní funkce, ketré nám zpřístupní nabídku se základními logickými funkcemi Booleovy algebry, například standartní logické bloky. Kliknutím na tlačítko Speciální funkce získáme výběr speciálních funkcí.

Obr. 17 Tlačítka funkcí

2.2.3 Vkládání bloků

Vložení bloků provádíme tak, že klikneme na skupinu ikon (Obr. 17), který obsahuje požadovaný blok. Všechny bloky vybrané skupiny jsou zobrazeny pod programovací plochou.

Obr. 18 Ukázka bloků

Kteroukoliv z těchto funkcí můžeme umístit na programovací plochu jednoduchým kliknutím myši. Vybereme si požadovaný blok a klikneme na něj levým tlačítkem myši. Pro vložení bloku klikneme opět levým tlačítkem myši na jakoukoliv část programovací plochy.

Bloky nemusíme hned vyrovnávat. Vyrovnat je můžeme, až když jich máme více, aby byl obvod přehlednější. Vyrovnání uděláme tak, že klikneme na ikonku Výběr na panelu nástrojů programování. A následně klikneme levým tlačítkem na blok, který chceme posunout a držíme levé tlačítko tak dlouho dokud blok neumístíme do potřebné polohy. Stav ikonky Výběr docílíme i stisknutím klávesy Escape na klávesnici.

2.2.4 Úpravy bloků

Úprava bloků slouží k jejich editaci a konfiguraci. Kliknutím pravého tlačítka vyvoláme kontextové menu, které nabízí různé možnosti editace objektu. Obsah tohoto kontextového menu závisí na vybraném objektu. Objekt zde není jen blok nebo spojovací čára, objektem rozumíme i programovací plochu a nástrojové lišty.

Konfiguraci bloků provádíme dvojitým kliknutím levého tlačítka myši. Tímto otevřeme okno, kde zadáváme vlastnosti bloku. Každý blok má své specifické vlastnosti, které můžeme měnit nebo mu určit.

2.2.5 Spojování bloků

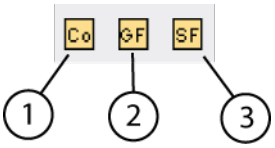
Aby byl obvod kompletní musíme všechny jednotlivé bloky vzájemně propojit. Učiníme tak pomocí volby tlačítka Spojit na panelu programovacích nástrojů.

Po kliknutí na toto tlačítko umístíme ukazatel myši na konektor bloku, kolem lterého se vysvítí modrý čtvereček. Stistkněme a podržíme levé tlačítko myší. Přemístíme ukazatel ze zdrojového konektoru k cílovému konektoru, u kterého se nám opět vysvítí modrý čtvereček. Uvolníme tlačítko myši. Takto spojíme dva bloky dohromady.

3 LOGICKÉ BLOKY

Logické bloky se člení do 3 skupin (obr. 19): Konstanty a konektory (1), základní funkce (2) a speciální funkce (3). Popíši zde logické bloky, které jsem použil v následujících navržených úlohách. Samozřejmě, že se jich v LOGO! Soft Comcort nachází mnohem více. Logické funkce jsou základem celého programu. Díky nim můžeme skládat různé typy úloh a příkladů.

Tato kapitola výchází z manuálu LOGO!

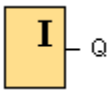


Obr. 19 Tři skupiny logických bloků

3.1 Konstanty a konektory

3.1.1 Vstupy

Symbol v LOGO!:



Obr. 20 Symbol vstupu

Výstupní bloky představují na modulu LOGO! výstupní svorky. Máme k dispozici až 24 digitálních vstupů.

V blokové konfiguraci můžeme přiřadit vstupnímu bloku novou vstupní svorku, pokud tato svorka není jižpoužívána v programu.

3.1.2 Výstupy

Symbol v LOGO!:

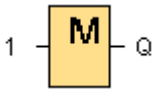
Obr. 21 Symbol výstupu

Výstupní bloky představují na modulu LOGO! výstupní svorky. Lze používat až 16 výstupů. Ve vaší blokové konfiguraci můžete přiřadit výstupnímu bloku novou svorky, pokud tato svorka není již používána v programu.

Výstup vždy přenáší signál z předchozího cyklu programu. Tato hodnota se během současného cyklu programu nemění.

3.1.3 Marker

Symbol l LOGO!:



Obr. 22 Symbol markeru

Výstupem markeru je jejich vstupní signál. LOGO! poskytuje 24 digitálních příznaků M1 – M24.

V blokové konfiguraci můžeme přiřadit markeru číslo, pokud toto číslo již v programu neexistuje.

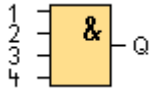
Výstup vždy přenáší signál z předchozího cyklu programu. Tato hodnota se během současného cyklu programu nemění.

3.2 Základní funkce

Základní funkce jsou jednoduché funkce Booleovi logiky. Budou zde uvedeny některé funkční bloky, které byli použity v navržených úlohách.

3.2.1 AND

Symbol v LOGO!:



Obr. 23 Symbol bloku AND

Výstupem funkce AND je 1 pouze jestliže všechny vstupy jsou 1, tj. Když jsou uzavřeny. Blokovému vstupu, který není používán (x), je přiřazena hodnota: x = 1.

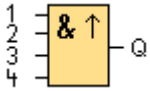
Logická tabula bloku AND:

Vstup 1	Vstup 2	Vstup 3	Vstup 4	Výstup
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Obr. 24 Logická tabulka bloku AND

3.2.2 AND s detekcí sestupné hrany

Symbol v LOGO!:

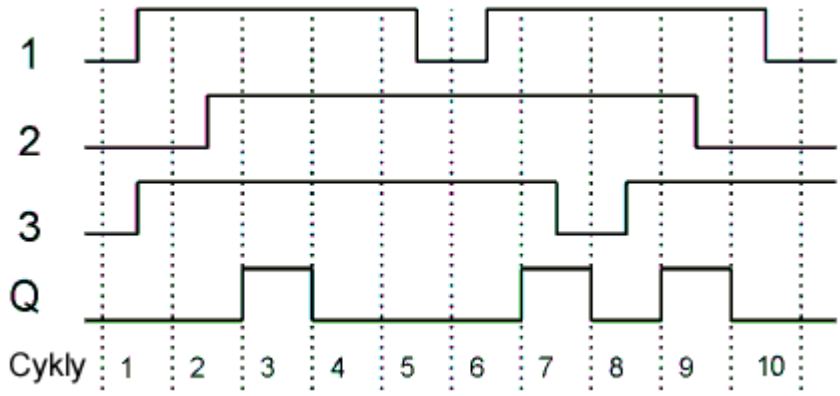


Obr. 25 Symbol bloku AND s detekcí sestupné hrany

Výstupem funkce AND s detekcí sestupné hrany je 1 pouze jestliže všechny vstupy jsou 1 a alespoň jeden vstup byl 0 během posledního cyklu.

Výstup je nastaven na 1 pro dobu jednoho cyklu a musí být resetován na 0 pro dobu příštího cyklu, než může být zase nastaven na 1.

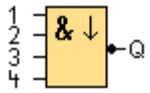
Blokovému vstupu, který není používán (x), je přiřazena hodnota: x = 1.



Obr. 26 Časový diagram funkce AND s detekcí sestupné hrany

3.2.3 NAND s detekcí sestupné hrany

Symbol v LOGO!:

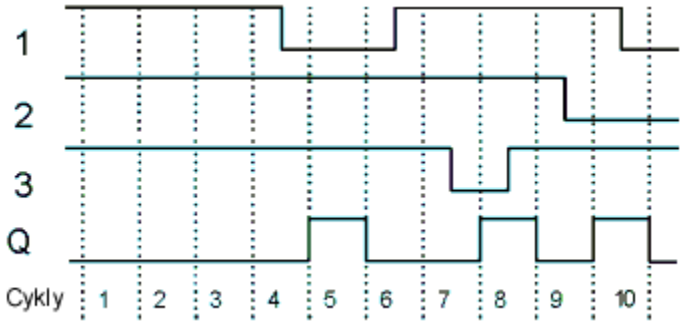


Obr. 27 Symbol bloku NAND s detekcí sestupné hrany

Výstupem funkce NAND s detekcí sestupné hrany je 1 pouze jestliže alespoň jeden vstup je 0 a všechny vstupy během posledního cyklu byly1.

Výstup je nastaven na 1 pro dobu jednoho cyklu a musí být resetován na 0 alespoň pro dobu příštího cyklu, než může být zase nastaven na 1.

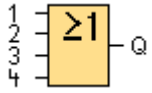
Blokovému vstupu, který není používán (x), je přiřazena hodnota: x = 1.



Obr. 28 Časový diagram funkce NAND s detekcí sestupné hrany

3.2.4 OR

Symbol v LOGO!:



Obr. 29 Symbol bloku OR

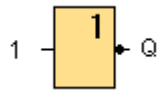
Výstupem funkce OR je 1, jestliže alespoň jeden výstup je 1, tj. když je uzavřený.
Blokovému vstupu, který není používán (x), je přiřazena hodnota: x = 1.
Logická tabulka funkce OR:

Vstup 1	Vstup 2	Vstup 3	Vstup 4	Výstup
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Obr. 30 Logická tabulka funkce OR

3.2.5 NOT

Symbol v LOGO:!



Obr. 31 Symbol bloku NOT

Výstupem je 1, jestliže vstup je 0. Blok funkce NOT převrací vstupní status.

Logická tabulka funkce NOT:

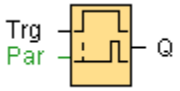
Vstup 1	Výstup
0	1
1	0

Obr. 32 Logická tabulka funkce NOT

3.3 Speciální funkce

3.3.1 Zpožděné zapnutí

Symbol v LOGO!:



Obr. 33 Symbol bloku zpožděného zapnutí

Výstup není zapnut, dokud nevyprší nastavená doba zpoždění.

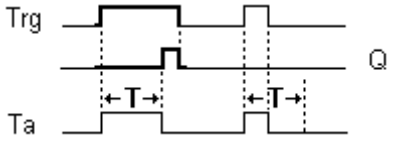
Připojení	Popis
Vstup **Trg**	Zpožděné zapnutí je spuštěno pomocí vstupu Trg (spouštěcí impulz).
Parametr	T je doba zpoždění, po níž bude zapnut výstup (výstupní signál se mění z 0 do 1). **Remanence** zapnuta = stav je uchován v paměti.
Výstup **Q**	Q zapíná po vypršení parametrizovaného času T, jestliže je ještě nastaven Trg.

Obr. 34 Tabulka s popisem zpožděného zapnutí

Parametrizovaný čas T může být vytvořen skutečnou hodnotou jiné, již naprogramované funkce:

- Analogové komparátory
- Analogové spínače
- Analogové zesilovače
- Dopředné a zpětné čítače

Požadovanou funkci vybereme pomocí čísla bloku.



Obr. 35 Časový diagram zpožděného zapnutí

Čas Ta (současný čas v LOGO!) je spuštěn s přechodem z 0 na 1 na vstupu Trg.

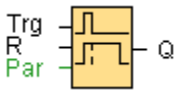
Zůstane-li stav vstupu Trg 1 alespoň po dobu nastaveného času T, je po uplynutí tohoto času výstup nastaven na 1 (signál zapnutí výstupu následuje signál zapnutí vstupu se zpožděním).

Čas je resetován, jestliže status na vstupu Trg se znovu změní na 0 před tím, než čas T vyprší.

Výstup je resetován na 0, když vstup Trg je 0.

3.3.2 Zpožděné vypnutí

Symbol v LOGO!:



Obr. 36 Symbol bloku zpožděného vypnutí

Výstup se zpožděným vypnutím není resetován, dokud definovaný čas nevyprší.

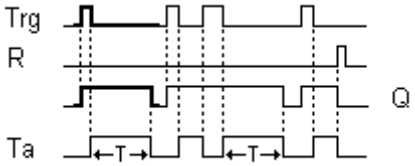
Připojení	Popis
Vstup **Trg**	Sestupná hrana A (změna z 1 do 0) na vstupu Trg (Trg znamená trigger = spouštěcí impulz) startuje čas pro zpožděné vypnutí.
Vstup **R**	Resetujte čas zpožděného vypnutí a nastavte výstup na 0 pomocí vstupu R (Reset). Reset má přednost před Trg
Parametr	T je doba zpoždění, po níž bude výstup vypnut (výstupní signál se změní z 1 do 0). **Remanence** zapnuta = stav je uchován v paměti.
Výstup **Q**	Q se zapíná po uplynutí času T po spuštění na vstupu Trg.

Obr. 37 Tabulka s popisem zpožděného vypnutí

Parametrizovaný čas T může být vytvořen skutečnou hodnotou jiné, již naprogramované funkce:

- Analogové komparátory
- Analogové spínače
- Analogové zesilovače
- Dopředné a zpětné čítače

Požadovanou funkci vybereme pomocí čísla bloku.



Obr. 38 Časový diagram zpožděného vypnutí

Výstup Q je nastaven na 1 současně s přechodem z 0 na 1 na vstupu Trg.

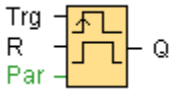
Při přechodu z 1 na 0 na vstupu Trg LOGO! Znovu spustí současný čas T a výstup zůstane nastaven. Výstup Q je resetován na 0, když Ta dosáhne hodnoty specifikované v T (Ta = T) (zpožděné vypnutí).

Impulz na vstupu Trg opětovně spustí čas Ta.

Můžeme resetovat řas Ta a výstup pomocí vstupu R (Reset) před tím, než čas Ta vypší

3.3.3 Hranou spouštěné relé

Symbol v LOGO!:

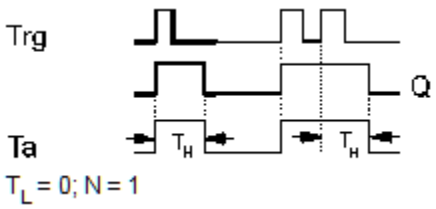


Obr. 39 Symbol bloku hranou spouštěné relé

Poté, co parametrizovaný čas zpoždění uplyne, generuje vstupní impulz nastavený počet výstupníchimpulzů s definovaným poměrem impulz/pauza (znovu spustitelných)

Připojení	Popis
Vstup **Trg**	Vstupem Trg (Trg znamená trigger = spouštěcí impulz) se spouští čas výstupního pulzu.
Vstup **R**	Výstup a současný čas Ta jsou resetovány na 0 při signálu na vstupu R.
Parametr	Mezipulzní doba T_L a pulzní doba T_H jsou nastavitelné. N určuje počet cyklů impulz/pauza T_L / T_H: Rozsah hodnot: 1...9. **Remanence** zapnuta = stav je uchován v paměti.
Výstup **Q**	Výstup Q je nastaven po uplynutí času T_L a resetován po uplynutí času T_H.

Obr. 40 Tabulka s popisem hranou spoštěného relé



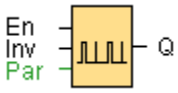
Obr. 41 Časový diagram hranou spouštěného relé

Se změnou na 1 na vstupu Trg je spuštěn čas T_L (nízký čas). Po uplynutí času T_L je výstup Q nastaven na 1 pro dobu trvání času T_H (vysoký čas).

Jestliže je vstup Trg znovu spuštěn před uplynutím nastaveného času (T_L + T_H), čas Ta je resetován a je opět spuštěna perioda impulz/pauza.

3.3.4 Asynchronní pulzní generátor

Symbol v LOGO!:



Obr. 42 Symbol bloku asynchronního pulzního generátoru

Tvar pulzu může být modifikován přes programovatelný poměr pulz/pauza.

Připojení	Popis
Vstup **En**	Vstupním signálem En můžete zapnout nebo vypnout asynchronní pulzní generátor.
Vstup **Inv**	Vstup Inv může být použit pro inverzi výstupního signálu aktivního asynchronního pulzního generátoru.
Parametr	**TL,TH:** Můžete přizpůsobit poměr mezi pulzem(TL) a pauzou(TH) ? definice střídy **Remanence** zapnuta = stav je uchován v paměti.
Výstup **Q**	Q se cyklicky přepíná s časy T_H a T_L.

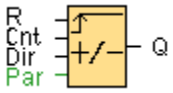
Obr. 43 Tabulka s popisem asynchronního pulzního generátoru

Můžeme nastavovat poměr pulzu/pauzy v parametrech TH (Time High) a TL (Time Low).

Vstup Inv může být použit pro invezi výstupního signálu. Vstup Inv invertuje výstup pouze tehdy, je-li blok zapnut pomocí En.

3.3.5 Dopředný a zpětný čítač

Symbol v LOGO!:

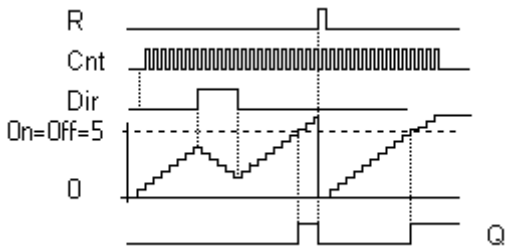


Obr. 44 Symbol bloku dopředného a zpětného čítače

Vstupní impulz zvyšuje nebo snižuje vnitřní hodnotu v závislosti na nastavení parametru. Výstup je nastaven nebo resetován, když je dosaženo kofigurovaného prahu. Směr počítání může být změněn signálem na vstupu Dir.

Připojení	Popis
Vstup **R**	Výstup a interní čítací hodnotu resetujete na nulu signálem na vstupu R (Reset).
Vstup **Cnt**	Tato funkce počítá změny z 0 na 1 na vstupu Cnt. Nepočítá změny z 1 na 0. Použijte ● ? Vstupy I5/I6 pro vysokofrekvenční počty (k dispozici pouze u určitých přístrojů LOGO!, viz manuál LOGO!): max. 2 kHz. ● ? Jakýkoli jiný vstup nebo prvek obvodu pro nižší frekvence (4 Hz).
Vstup **Dir**	Vstup Dir (Direction) určuje směr počítání: Dir = 0: čítání vpřed Dir = 1: Nazpět
Parametr	Zapnut: Práh zapnutí Rozsah hodnot: 0...999999 Vypnut: Práh vypnutí Rozsah hodnot: 0...999999 **Remanence** zapnuta = stav je uchován v paměti.
Výstup **Q**	Q je nastavován a resetován podle skutečné hodnoty na Cnt a nastavených prahů.

Obr. 45 Tabulka s popisem dopředného a zpětného čítače



Obr. 46 Časový diagram dopředného a zpětného čítače

Při každé pozitivní hraně na vstupu Cnt je interní čítač zvýšen o jedničku (Dir = 0) nebo snížen o jedničku (Dir = 1).

Hodnotu vnitřního čítače můžete resetovat na 0 signálem na vstupu resetu R. Dokud R = 1, je výstup 0 a pulzy na vstupu Cnt se nepočítají.

Výstup Q je nastavován a resetován podle skutečné hodnoty na Cnt a nastavených prahů. Viz následující pravidla pro výpočty.

Pravidlo pro výpočet:

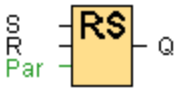
- Jestliže práh zapnutí >= práh vypnutí, pak:
  Q = 1, jestliže Cnt >= On
  Q = 0, jestliže Cnt < Off
- Jestliže práh zapnutí < práh vypnutí, pak:
  Q = 1, jestliže On <= Cnt < Off

Nastavené mezní hodnoty pro parametry zapnutí a nebo vypnutí mohou být také odvozeny od skutečné hodnoty jiné, již konfiigurované funkce. Můžete použít skutečnou hodnotu následujících funkcí:

- Analogový komparátor (skutečná hodnota Ax, Ay)
- Analogový spínač (skutečná hodnota Ax)
- Analogový zesilovač (skutečná hodnota Ax)
- Vzestupný/sestupný čítač (skutečná hodnota Cnt)

3.3.6 Samodržné relé

Symbol v LOGO!:

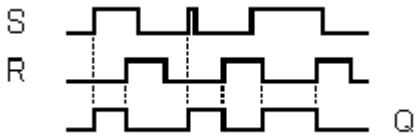


Obr. 47 Symbol bloku samodržného relé

Vstup S nastavuje výstup Q na 1 a další vstup R nuluje výstup.

Připojení	Popis
Vstup **S**	Vstup S (Set) nastaví výstup Q na 1.
Vstup **R**	Vstup R (Reset) znovu nastaví výstup Q na 1. Výstup Q je resetován, jestliže S a R jsou oba nastaveny (resetování má přednost před nastavováním).
Parametr	**Remanence** zapnuta = stav je uchován v paměti.
Výstup **Q**	Q je nastaven při signálu na vstupu S a zůstává nastaven, dokud není resetován signálem na vstupu R.

Obr. 48 Tabulka s popisem samodržného relé



Obr. 49 Časový diagram samodržného relé

Samodržné relé je jednoduchý binární paměťový prvek. Signál na výstupu závisí na stavu vstupů a na předchozím stavu výstupu.

Logická tabulka samodržného relé:

S	R	Q	Poznámka
0	0	-	Status nezměněn
0	1	0	Nulování
1	0	1	Nastavení
1	1	0	Nulování

Obr. 50 Logická tabulka samodržného relé

4 NAVRŽENÉ ÚLOHY

Zde jsou vypracovány a popsány dvě navržené úlohy, na kterých jsou použity základní funkční bloky programu LOGO! Soft Comfort. První úloha je vypracována ve dvou řešení. Druhá úloha je rozdělena do dvou částí. V úlohách jsou použity v praxi pravdivostní tabulky a Karnaughovy mapy.

4.1 Semafor

První ukázková úloha je obyčejný semafor, který můžeme vidět v běžném provozu na komunikacích. Semafor má 3 barevně odlišné světla: zelenou, oranžovou a červenou a každá barva může být ve stavu buď zapnutém nebo vypnutém. Chceme tedy aby to splňovalo následující požadavky:

1) Ve vypnutém režimu chceme aby blikalo pouze oranžové světlo v intervalu po půl sekundách.
2) V zapnutém režimu chceme aby nám červená svítila 10 sekund, poté potřebujeme abychom se nám na 2 sekundy rozsvítilo oranžové světlo současně s červeným světlem. Po tomto procesu je potřeba aby se zelené světlo rozsvítilo na 10 sekund. Po zhasnutí zeleného světla chceme aby se rozsvítilo oranžové světlo na 2 sekundy a poté aby se opět rozsvítilo světlo červené na 10 sekund a tento celý děj aby se opakoval neustále dokola.

Úlohu vypracujeme ve dvou možných řešení. Pomocí dopředných a zpětných čítačů a nebo pomocí pulzních relé.

Obr. 51 Semafor

Pro obě dvě řešení je základ stejný, a to je funkce semaforu. Zjistíme si ji za pomoci pravdivostní tabulky a Karnaughovy mapy. Při tvoření pravdivostní tabulky dávame pozor na to, aby semafor splňoval dané požadavky. V tomto případě má semafor 4 vstupy a 3 výstupy.

Vstupy jsou požadované funkce semaforu a ty jsou :

1. Červené světlo svítí (RED)
2. Červené s oranžovým světlem svítí (RED + ORANGE)
3. Oranžové světlo svítí (ORANGE)
4. Zelené světlo svítí (GREEN)

Výstupy tvoří tři světla:

1. Červené světlo (RED LIGHT)
2. Oranžové světlo (ORANGE LIGHT)
3. Zelené světlo (GREEN LIGHT)

Pravdivostní tabulka:

GREEN	OR.	RED + OR.	RED	GREEN L.	ORANGE L.	RED L.
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	0	1	1	0	1	1
0	1	0	0	0	1	0
0	1	0	1	0	1	1
0	1	1	0	0	1	1
0	1	1	1	0	1	1
1	0	0	0	1	0	0
1	0	0	1	1	0	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	0
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

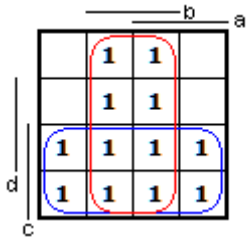
Obr. 52 Pravdivostní tabulka semaforu

Podle této pravdivostní tabulky vytvoříme Karnaughovy mapy pro výstupy, které budou vypadat následovně:

Pro proměnné v Karnaughových mapách platí:

a = GREEN
b = ORANGE
c = RED + ORANGE
d = RED

ORANGE LIGHT:

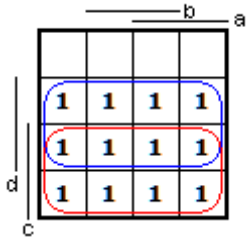


Obr. 53 Karnaughova mapa pro výstup ORANGE LIGHT

Z této Karnaughovy mapy platí:

ORANGE LIGHT = ORANGE + (RED + ORANGE)

RED LIGHT:

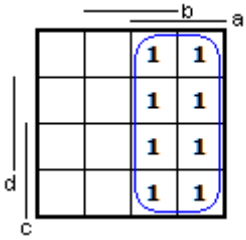


Obr. 54 Karnaughova mapa pro výstup RED LIGHT

Z této Karnaughovy mapy platí:

RED LIGHT = RED + (RED + ORANGE)

GREEN LIGHT:



Obr. 55 Karnaughova mapa pro výstup GREEN LIGHT

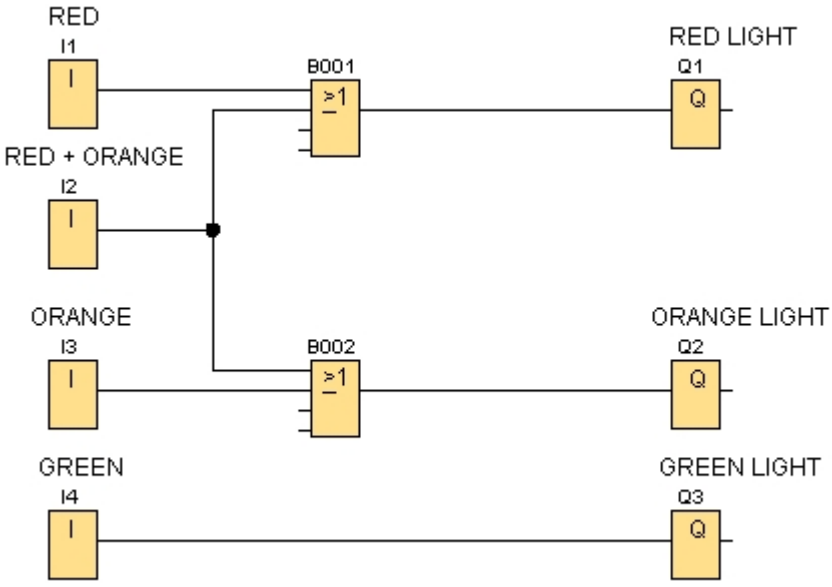
Z této Karnaughovy mapy platí:

GREEN LIGHT = GREEN

Ze všech Karnaughových map jsme získaly tyto rovnice:

RED LIGHT = RED + (RED + ORANGE)
ORANGE LIGHT = ORANGE + (RED + ORANGE)
GREEN LIGHT = GREEN

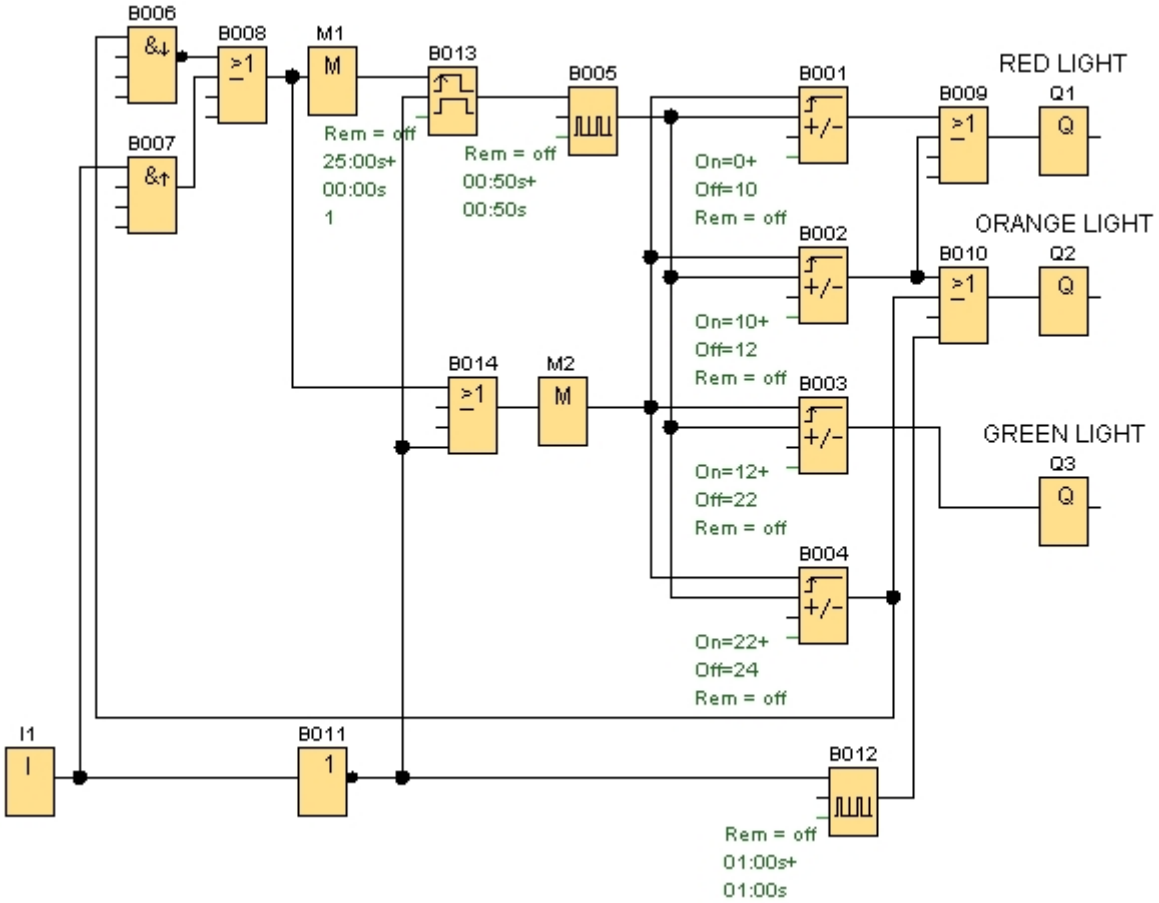
Z těchto ronic vznikne následující základní funkce a schema semaforu:



Obr. 56 Schema základní funkce semaforu

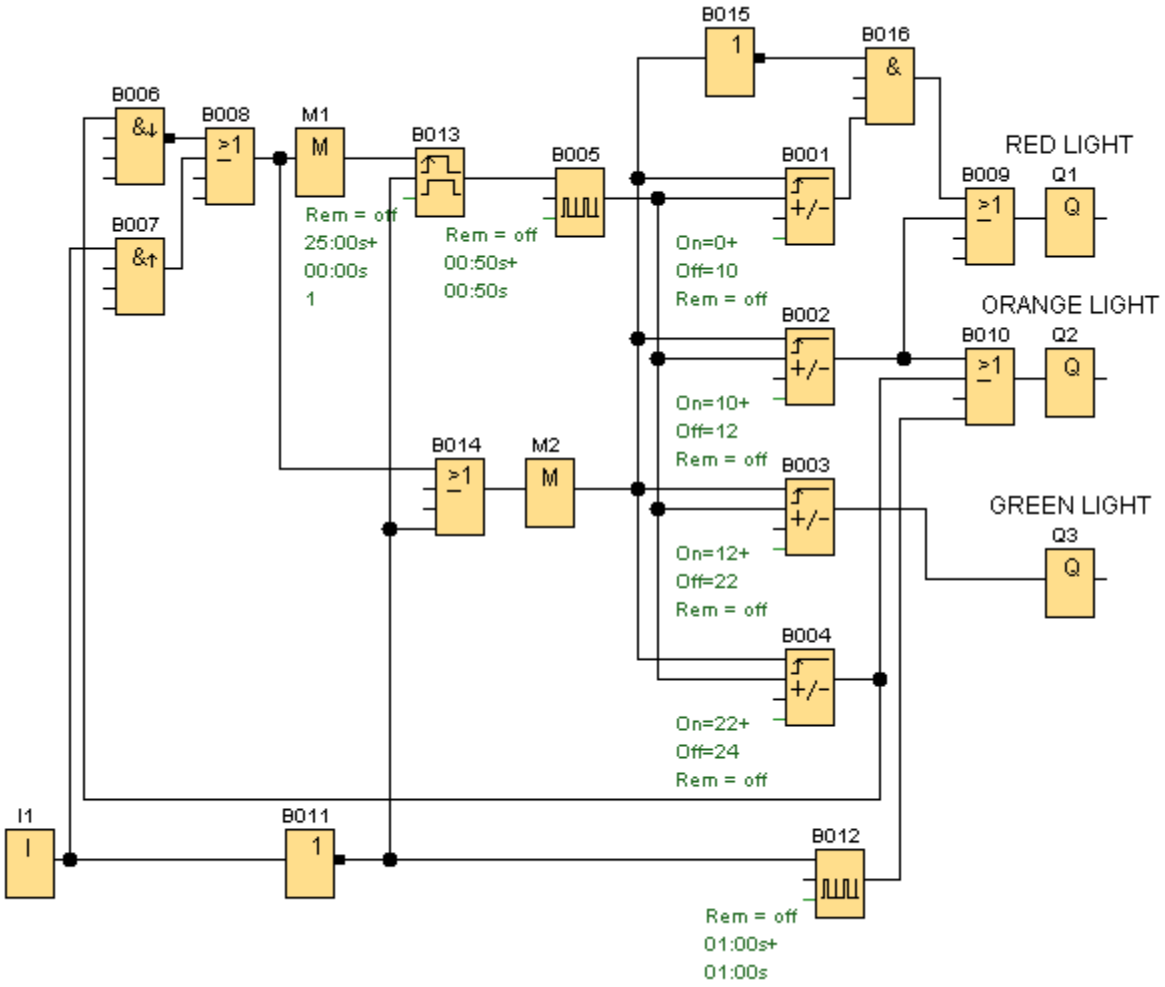
4.1.1 Pomocí čítačů

Zde je úloha vyřešena pomocí dopředných a zpětných čítačů. Funguje na principu generování pulzů asynchronním pulzním generátorem. Ten se dá nastavit na libovolný časový interval mezi pulzy. Tyto pulzy pak zachytává dopředný a zpětný čítač a počítá je. Při dosažení nastaveného počtu pulzů změní svou logickou hodnotu z 0 na 1 a po dosažení dalšího nastaveného limitu se vypne. Tím se postupně naplňuje hodnota všech čítačů. Každý čítač je nastaven na jiný počet impulzů a tak postupně zapínají a vypínají ty světla která potřebujeme. Pro opakovaný cyklus je potřeba vždy tyto čítače resetovat.



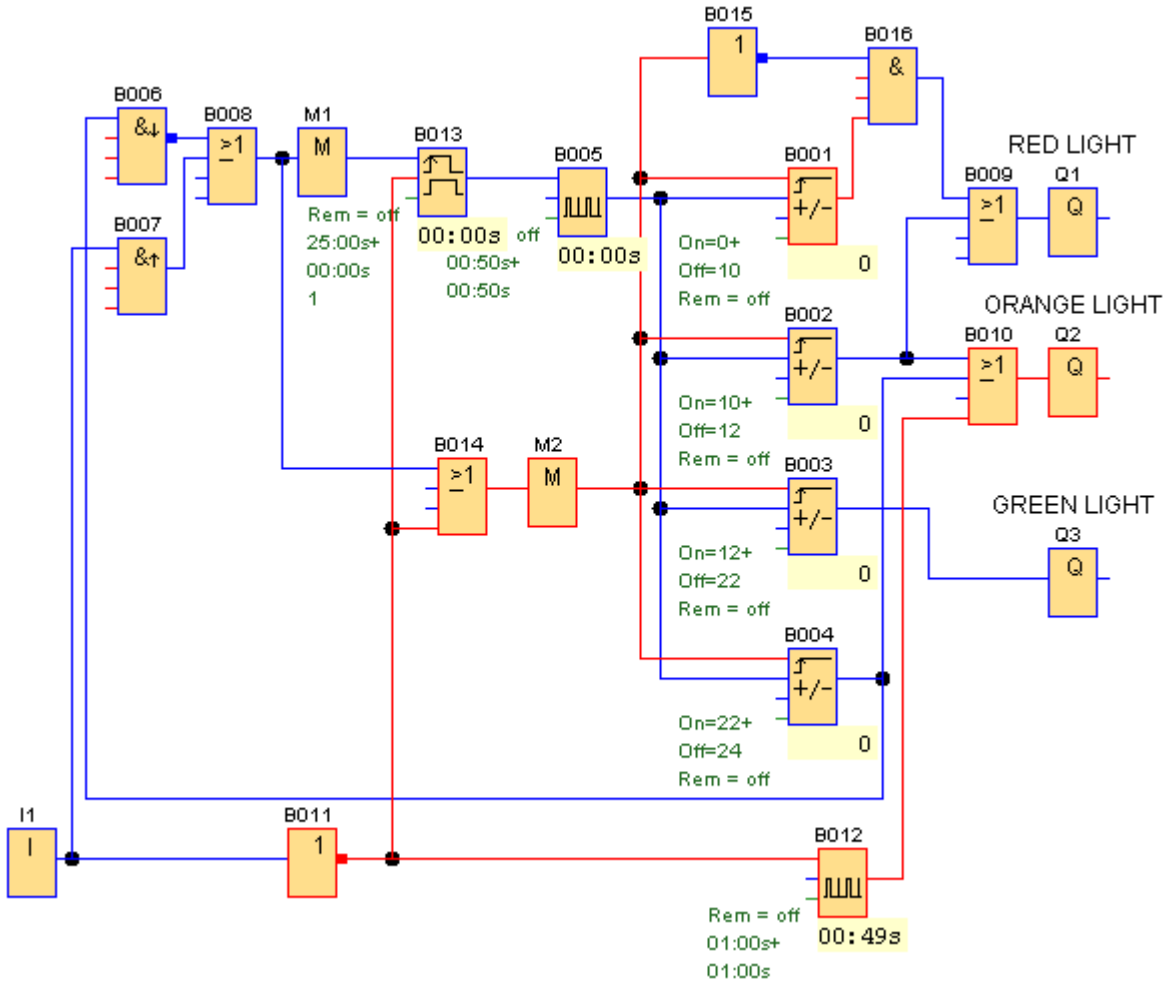
Obr. 57 Zapojení semaforu pomocí dopředných a zpětných čítačů ve verzi 0BA4

Ve verzi LOGO! 0BA4, ve které se původně vypracovávala tato úloha, fungovalo předchozí schéma (obr. 57) bez problémů. Po získání nové verze 0BA5 a následném odzkoušení v ní, se zjistilo, že schéma nepracuje jak má. Při vypnutí semaforu, kdy má blikat pouze oranžové světlo, do toho svítilo i červené světlo. Bylo to způsobeno blokem dopředného a zpětného čítače (B001), který je nastaven tak, aby se zapnul při nultém pulzu. Při vypnutí semaforu je ale na čítače přiveden reset, což u starší verze mělo za následek vyresetování čítače a jeho nefunkčnost. U nové verze čítač funguje i když je přiveden signál na reset. Proto se musela udělat malá změna aby se zachovala funkčnost semaforu.



Obr. 58 Zapojení semaforu pomocí dopředných a zpětných čítačů ve verzi 0BA5

Zapnutí nebo vypnutí úlohy řešíme vypínačem (I1). Po sepnutí vypínače přivedeme logickou hodnotu 1 na NAND s detekcí náběžné hrany (B007) a tím dostane, přes logický blok OR (B008), spouštěcí signál pro blok hranou spouštěné relé (B013), které nám drží celý cyklus zapnutý (viz. níže). Pokud je vypínač vypnutý, tak nám díky logickému bloku NOT (B011), který neguje logickou hodnotu, přivádí signál na reset hranou spouštěného relé a přes logický blok OR (B014) na reset dopředných a zpětných čítačů (B001 – B004). Současně spouští asynchronní pulzní generátor (B012), který je navedený na oranžové světlo. Ten má nastavenu šířku pulzu na 1 sekundu a délku pauzy také na 1 sekundu. Tím docílíme blikání oranžového světla.



Obr. 59 Vypnutý semafor pomocí dopředných a zpětných čítačů

Funkčnost je zajištěna pomocí hranou spouštěného relé (B013), asynchronního pulzního generátoru (B005), bloku NOT (B015), bloku AND (B016) a dopředných a zpětných čítačů (B001 – B004).

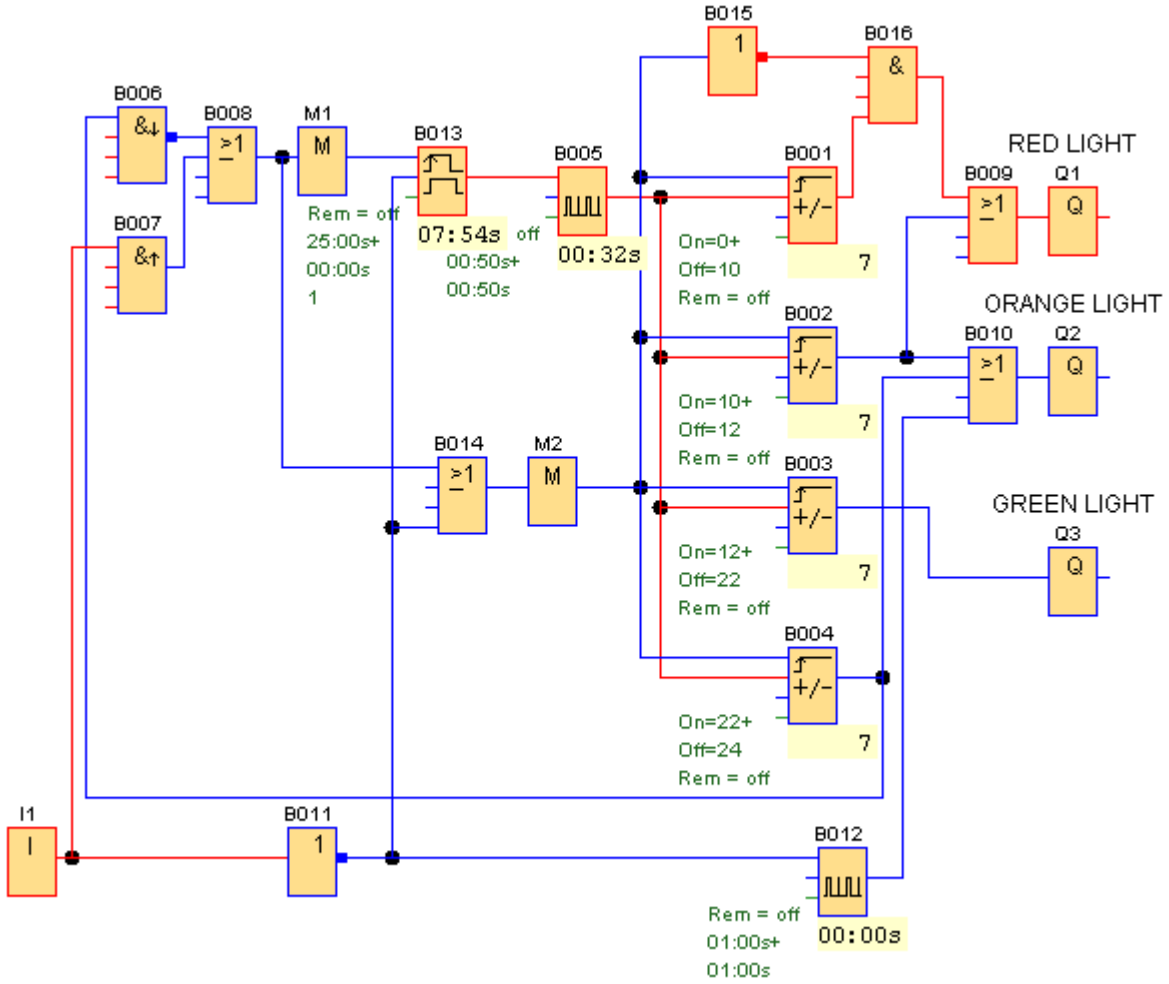
Hranou spouštěné relé udržuje asynchronní pulzní generátor na logické hodnotě 1. Kdyby tomu tak nebylo, generátor by nefungoval, proto musí být toto relé nastavené na delší dobu než je celý cyklus úlohy, to je např. 25 sekund.

Generátor má za úkol generovat pulzy v časovém intervalu 0,5 sekundy, kdy půl sekundy má logickou hodnotu 1 a půl sekundy logickou hodnotu 0. Tím zajistíme každý puls po 1 sekundě. Čítače počítají tyto pulzy a podle jejich počtu se zapínají a vypínají. Každý samozřejmě jinak.

Čítač který je na zapnutí a vypnutí červeného světla (B001) má zapínací práh nastaven na nultém pulzu, tedy okamžitě při zapnutí semaforu je čítač sepnutý. Aby mohl být sepnutý po dobu 10ti sekund, je jeho vypínací práh nastaven na 10 pulzů. Nastane ale problém při vypnutém semaforu, kdy má blikat pouze oranžové světlo. U čítače je přiveden signál na reset, ale čítač je zapnutý, protože je zde zapínací práh nastaven na nultý pulz a červené světlo je tudíž zapnuté. Musíme proto použít blok NOT (B015) a blok AND (B016) abychom docílily požadovaného výsledku. Signál, který nám resetuje čítače přivedeme na blok NOT (B015), tím ho znegujeme a tento znegovaný signál dále vstupuje do bloku AND (B016). Současně do tohoto bloku AND (B016) vstupuje signál z čítače (B001). Z bloku AND (B001) vystupuje signál pro zapnutí nebo vypnutí červeného světla. Dosáhneme tím požadovanou funkčnost. To je, že se nám červené světlo rozsvítí, jen pokud nejsou resetovány

čítače a je-li přiveden signál pro zapnutí červeného světla.

Ostatní čítače nastavíme dle zadané úlohy ve sledu pulzů, tzn. že u druhého čítače, který má sepnout oranžové světlo na 2 sekundy (B002), nastavíme jeho zapínací práh na 10 pulzů a vypínací práh na 12 pulzů. Následné pulzy musíme k dalším čítačům přičítat.

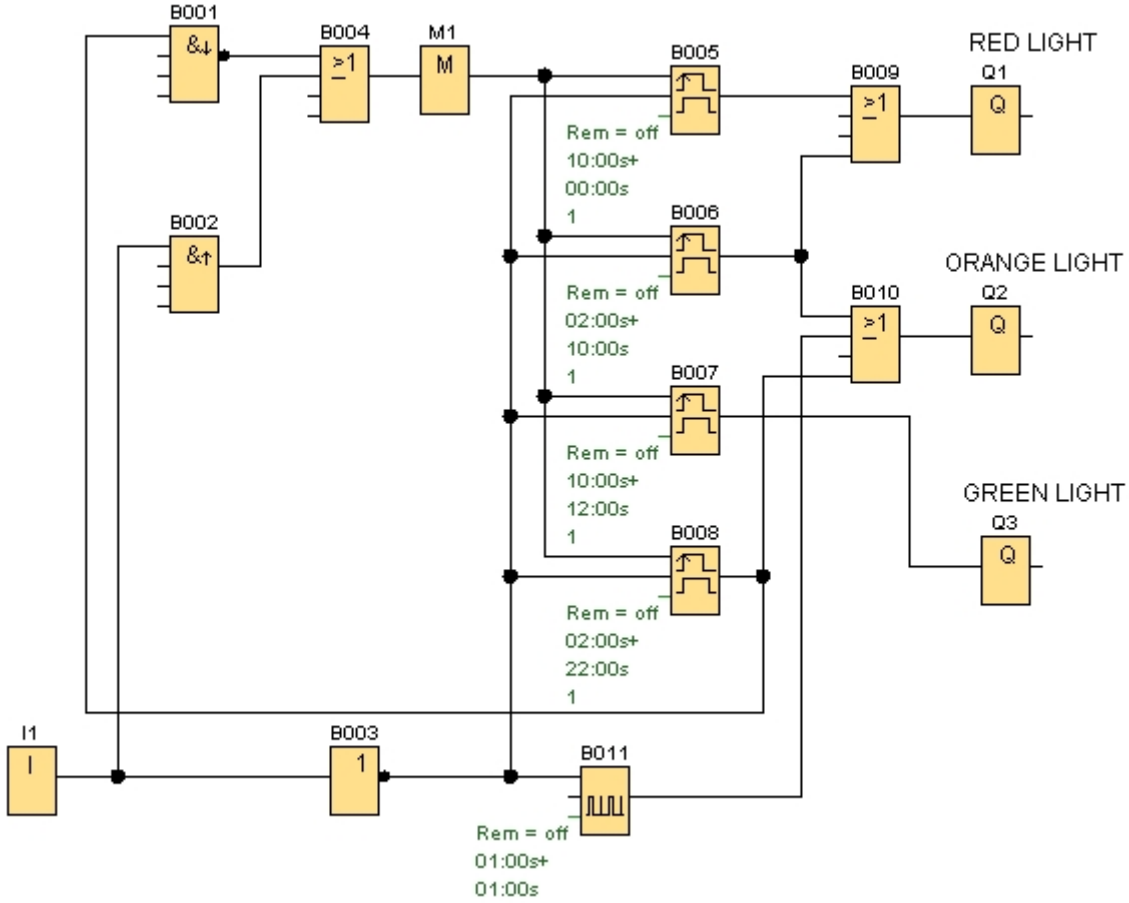


Obr. 60 Zapnutý semafor pomocí dopředných a zpětných čítačů

Zpětná větev je řešena tak, že až se poslední čítač (B004) naplní na nastavený počet pulzů, což je 24, přepne svou logickou hodnotu z 1 na 0 a tato logická hodnota se nám dostane i na blok NAND s detekcí sestupné hrany (B006) a tímto vygeneruje pulz o logické hodnotě 1, který sepne hranou spouštěné relé (B013) a resetuje dopředné a zpětné čítače (B001 – B004) a celý proces se může opakovat znovu.

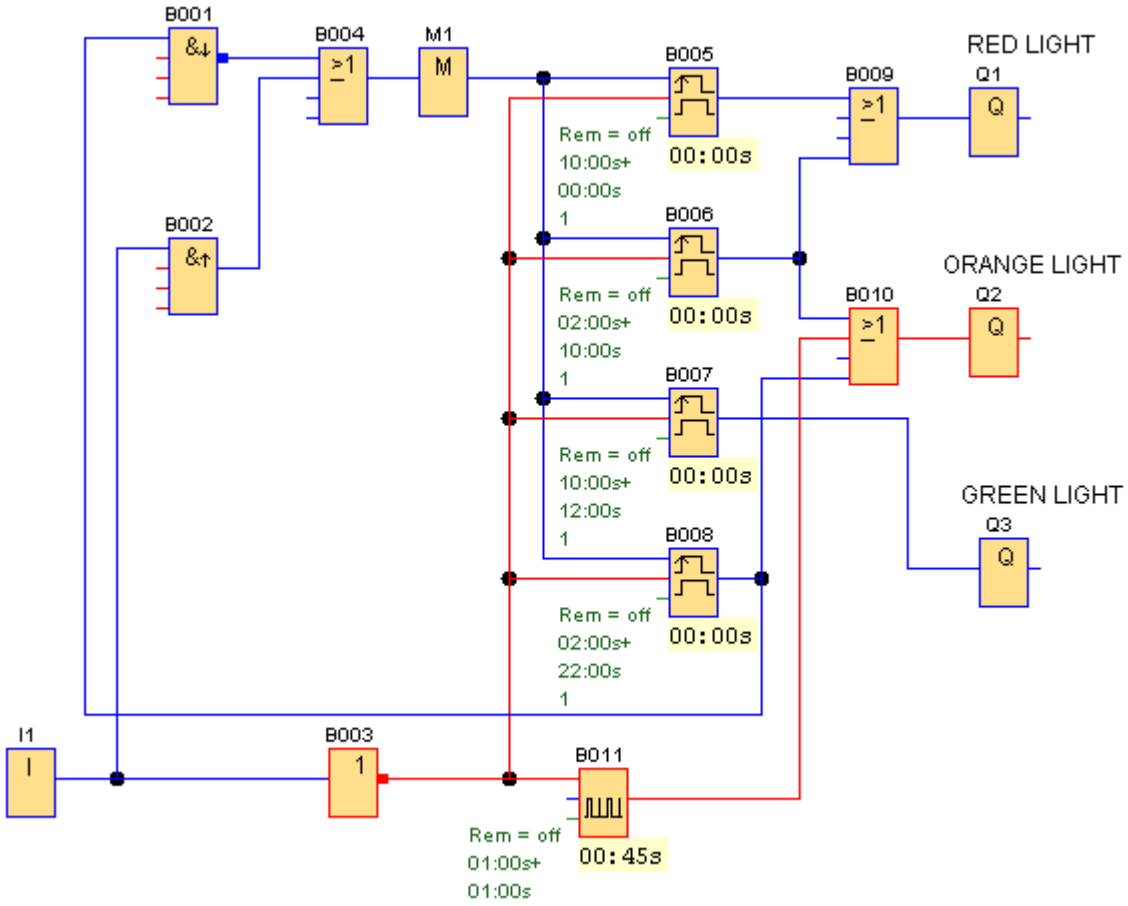
4.1.2 Pomocí pulzních relé

Úlohu řešíme pomocí pulzního relé. To pracuje na principu odpočítávání. Nejprve musíme na pulzní relé přivézt puls, který má logickou hodnotu 1. Poté se pulzní relé sepne a začne se odpočítávat nastavený čas. Po uplynutí této doby se změní logická hodnota z 0 na 1. Každé pulzní relé je zde nastaveno na rozdílný čas a zapíná tím potřebná světla. Pro opakování tohoto cyklu se musí pulzní relé resetovat.



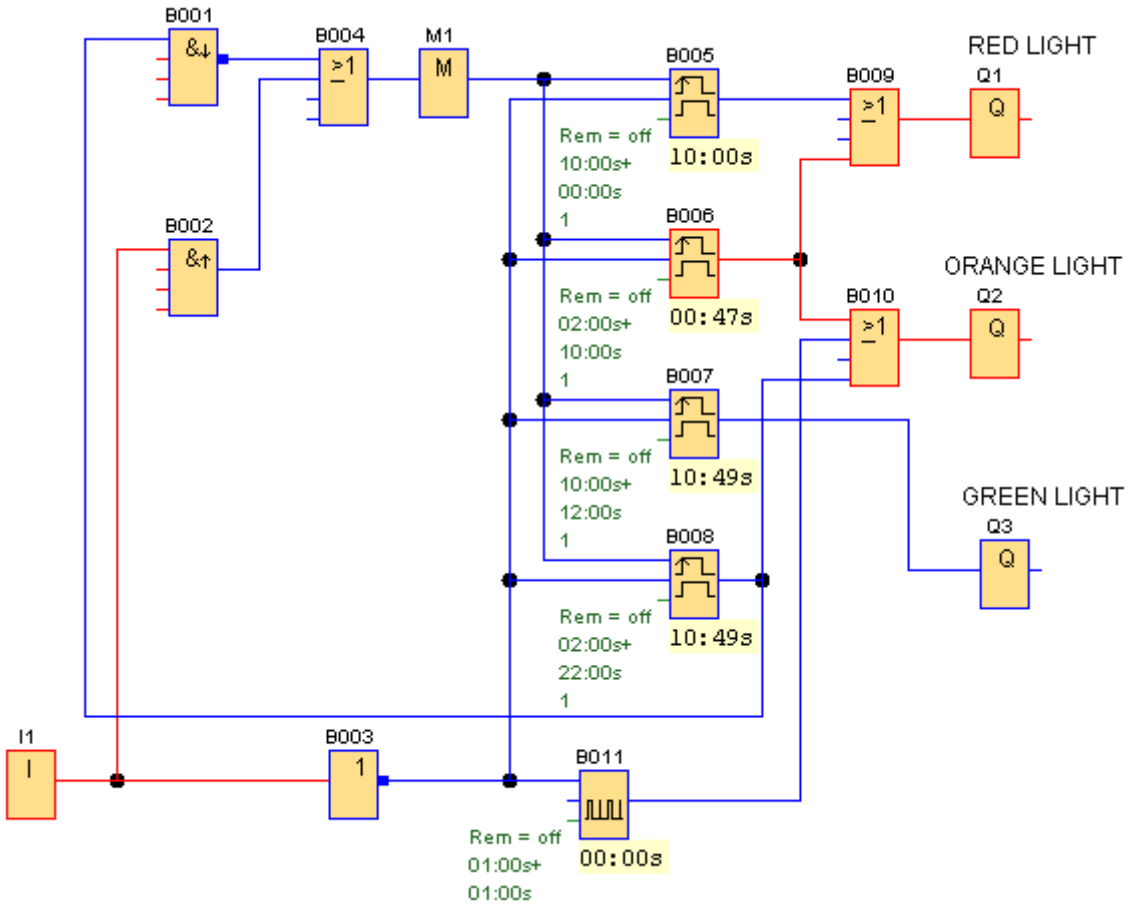
Obr. 61 Zapojení semaforu pomocí hranou spouštěných relé

Zapnutí nebo vypnutí úlohy řešíme vypínačem (I1). Po sepnutí vypínače přivedeme logickou hodnotu 1 na NAND s detekcí náběžné hrany (B002) a tím zapne, přes logický blok OR (B004), všechny hranou spouštěné relé. Tyto relé se nám starají o celou funkci semaforu (viz. níže). Pokud je vypínač vypnutý, tak nám díky logickému bloku NOT (B003), který neguje logickou hodnotu, přivádí signál na reset všech hranou spouštěného relé. Současně spouští asynchronní pulzní generátor (B011), který je navedený na oranžové světlo. Ten má nastavenu šířku pulzu na 1 sekundu a délku pauzy také na 1 sekundu. Tím docílíme blikání oranžového světla.



Obr. 62 Vypnutý semafor pomocí hranou spouštěných relé

Funkci semaforu řešíme za pomoci hranou spouštěných relé (B005 – B008). Každé relé má nastavenou jinou šířku pulzu a jinou mezipulzní šířku v závislosti na tom jaké světlo se má kdy zapnout. První relé, které je u červeného světla (B005) má nastavenu šířku pulzu na 10 sekund a mezipulzní šířku na 0 sekund, tj. že se hned po zapnutí vypínače rozsvítí. U druhého relé, které má sepnout oranžové světlo s červeným (B006), se nastaví šířka pulzu na 2 sekundy a mezipulzní šířka na 10 sekund. Následně se takhle nastaví všechna relé, dle zadání úlohy. Všechny relé běží současně, musí se zde tudíž dopočítat kdy se má sepnout další relé, aby se rozsvítila požadovaná světla.



Obr. 63 Zapnutý semafor pomocí hranou spouštěných relé

Zpětná větev je řešena tak, že až poslední hranou spouštěné relé (B008) přepne svou logickou hodnotu z 1 na 0, současně se tak přepne i blok NAND s detekcí sestupné hrany (B002) a tímto vygeneruje pulz o logické hodnotě 1, který sepne všechny hranou spouštěné relé (B005 - B008) a celý proces se opakje znovu.

4.2 Železniční přejezd

Druhá ukázková úloha je semafor se závorou na železničním přejezdu. Tato úloha má 2 části.

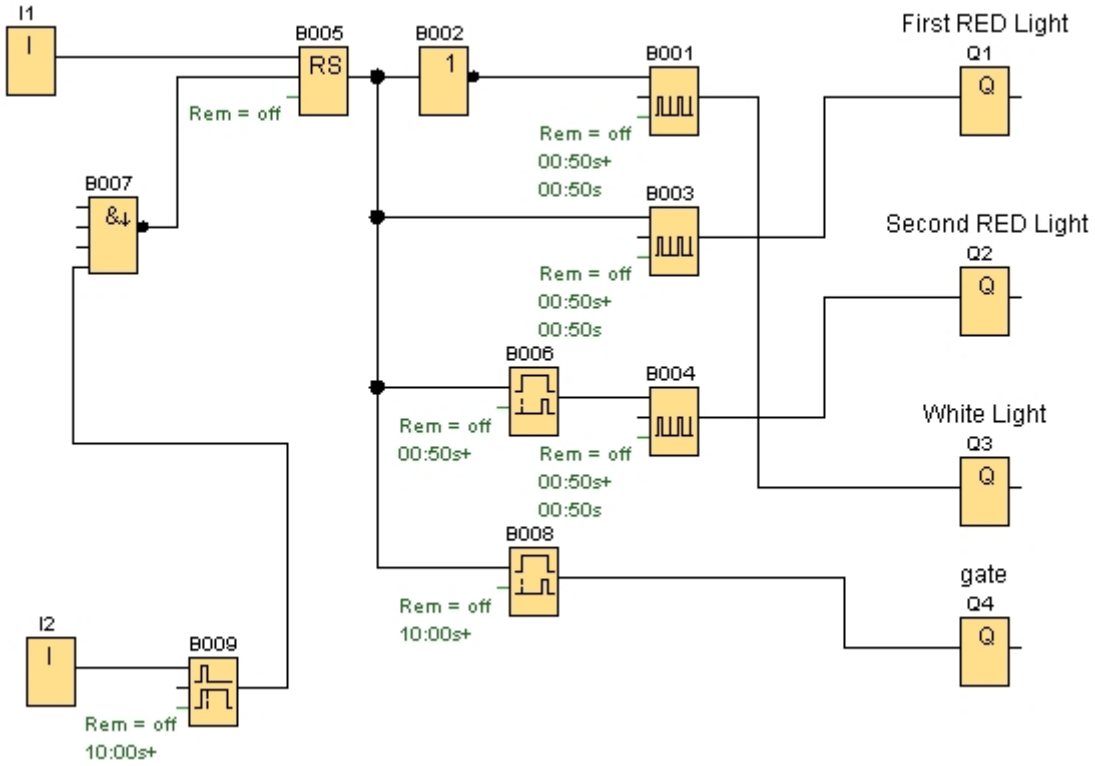
První část úlohy: Semafor má 3 barevně odlišné světla: jedno bílé světlo, a dvě červená světla. Na kolejích se nachází dva spínače, které zaznamenají příjezd a odjezd vlaku ze stanice. Dokud vlak nepřijede do stanice a nesepne první spínač, tak na semaforu bliká bílé světlo v intervalu po půl sekundách. Až se vlak blíží ke stanici, tak sepne první spínač, který sepne obě červená světla tak, aby blikala střídavě po půl sekundách a za 10 sekund se začne zavírat závora. Když vlak bude opouštět stanici a sepne druhý spínač, tak předchozí stav bude zachován a změna nastane až když poslední vagón vlaku rozepne druhý spínač a to taková změna, že po 10ti sekundách přestanou blikat červená světla a začne blikat bílé světlo v opět stejném intervalu jak v předchozím stavu což je půl sekundy a zvedne se závora.

Druhá část úlohy: Chceme aby závora měla dva koncové spínače, která budou snímat jestli je závora v horní poloze nebo spodní a podle toho požadujeme, aby se motor roztačel takovým směrem, aby šla závora nahoru, dolů a nebo motor stál.

Obr. 64 Fotka železničního přejezdu se závorami

4.2.1 První část úlohy

Tuto úlohu řešíme pomocí asynchronních pulzních generátorů, zpožděného zapnutí, zpožděného vypnutí a samodržného relé. Na blikající červená světla a na blikající bíle světlo použijeme asynchronní pulzní generátor. U něj si nastavíme pouze prodlevy mezi pulzy a u jednoho červeného světla je nastaveno zpožděné zapnutí o takovou časovou prodlevu, jaká je prodleva u jednoho impulzu. Zpožděné zapnutí musíme použít i u spouštění závory. Zpožděné vypnutí použijeme u zvedání závory a u vypínání blikání červených světel. Samodržné relé je zde proto, protože drží logickou hodnotu 1 i když už vlak nemá sepnuté čidlo kterým se zapínají výstražné znamení. To se vypne až když se samodržné relé resetuje.



Obr. 65 Schéma zapojení železničního přejezdu se závorou

4.2.2 Druhá část úlohy

Úlohu řešíme za pomocí pravdivostní tabulky a Karnaughovy mapy. Při tvoření pravdivostní tabulky dbáme na to abychom zachovaly správnou funkčnost a určily správně všechny vstupy a výstupy. V tomto případě máme 3 vstupy a 2 výstupy.

Vstupy jsou požadované funkce dle zadání:

1. Signál jestli má být závora otevřená nebo zavřená, 0 = otevřená závora, 1 = zavřená závora (GATE)
2. Horní koncový spínač (KSN)
3. Dolní koncový spínač (KSD)

Výstupy tvoří stavy motoru:

1. Motor zvedá závoru nahoru (MN)
2. Motor spouští závoru dolů (MD)

Pravdivostní tabulka:

GATE	KSN	KSD	MN	MD
0	0	0	1	0
1	0	0	0	1
0	1	0	0	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	0
0	1	1	-	-
1	1	1	-	-

Obr. 66 Celá pravdivostní tabulka koncových spínačů semaforu

V pravdivostní tabulce vidíme 2 stavy které nemohou nastat, to je když je horní i spodní spínač sepnutý, proto je odstraníme a tím nám vznikne tato pravdivostní tabulka:

GATE	KSN	KSD	MN	MD
0	0	0	1	0
1	0	0	0	1
0	1	0	0	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	0

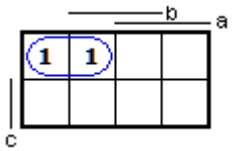
Obr. 67 Konečná pravdivostní tabulka koncových spínačů semaforu

Podle této pravdivostní tabulky vytvoříme Karnaughovy mapy pro výstupy, které budou vypadat následovně:

Pro proměnné v Karnaughových mapách platí:

a = KSD
b = KSN
c = GATE

MN:

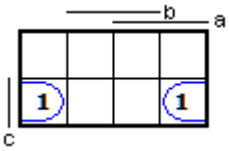


Obr. 68 Karnaughova mapa pro výstup MN

Z této Karnaughovy mapy platí:

$$MN = \overline{SIGNAL} \cdot \overline{KSN}$$

MD:



Obr. 69 Karnaughova mapa pro výstup MD

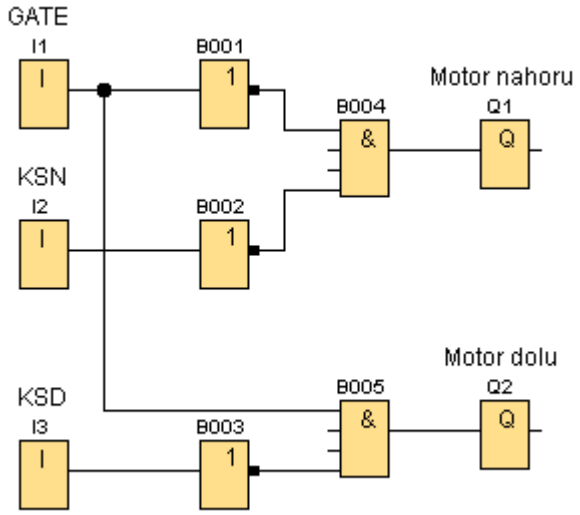
Z této Karnaughovy mapy platí:

$$MD = SIGNAL \cdot \overline{KSD}$$

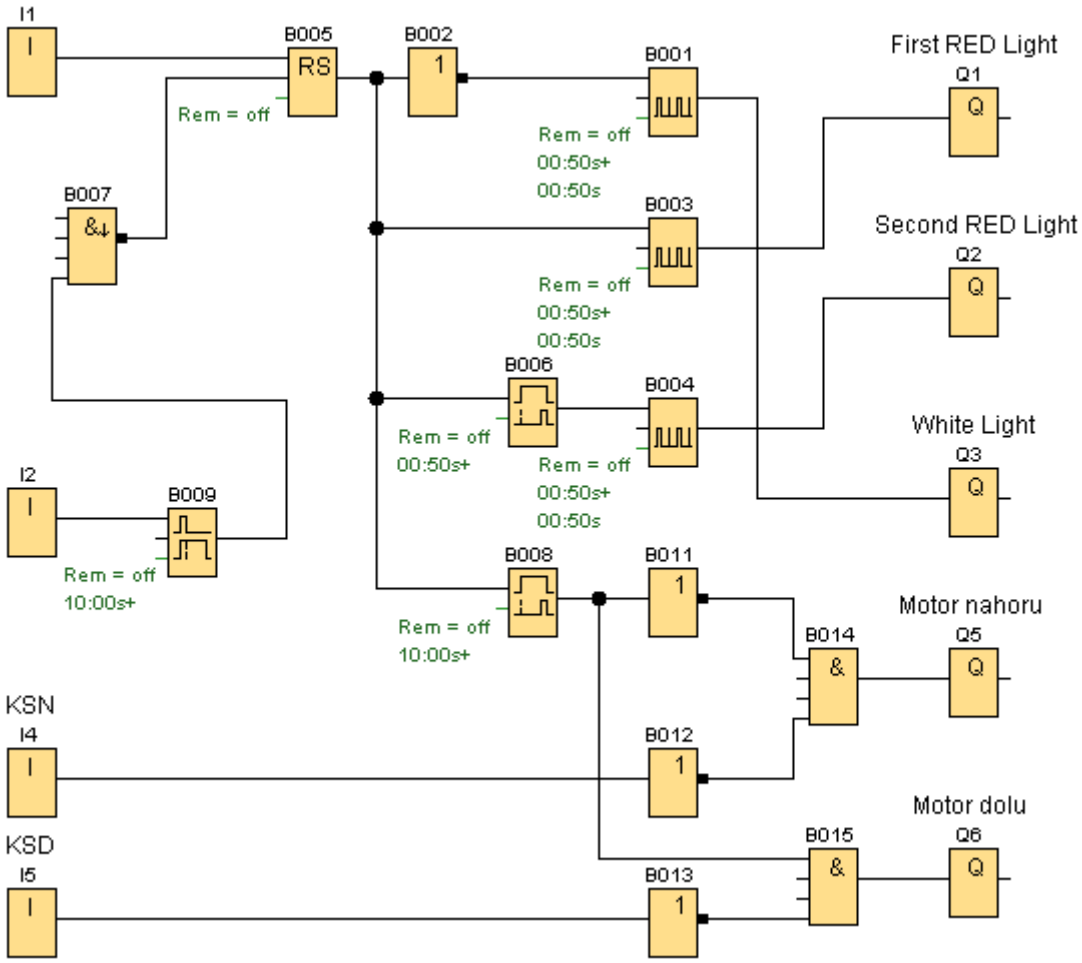
Z obou Karnaughových map jsme získaly tyto rovnice:

$$MN = \overline{SIGNAL} \cdot \overline{KSN}$$
$$MD = SIGNAL \cdot \overline{KSD}$$

Z těchto rovnic vznikne následujícíschéma pro Koncové spínače semaforu:



Obr. 70 Schema zapojení koncových spínačů u závory



Obr. 71 Celé schema železničního přejezdu se závorou a s koncovými spínači

5 ZÁVĚR

Po seznámení se s tímto programovacím softwarem a jeho funkčními bloky vznikla tato práce pro studenty s tím, aby jim pomohla lépe pochopit tuto problematiku. Bylo popsáno uživatelské rozhraní, jeho základy pro práci v tomto programu a také jsou zde vysvětleny funkční bloky, které se dále použily v navržených příkladech. Pro ukázku využití tohoto sotwaru a použití jeho funkčních bloků v praxi, vznikly navržené příklady, které jsou popsány a můžeme zde vidět funkci těchto bloků. Tato práce jako celek vznikla pro potřeby předmětu logického řízení a programovatelné automaty.

SEZNAM POUŽITÉ LITERATURY

[1] Poliščuk, R.: Titulní strana závěrečné práce, ,
[2] Poliščuk, R.: Instrukce pro autory závěrečných prací, 2006, http://autnt.fme.vutbr.cz/doc/SZZ2006_Instrukce.pdf
[3] Firemní materiály o LOGO! firmy SIEMENS
[4] LOGO! 0BA5: manuál, http://www1.siemens.cz/ad/current/content/data_files/automatizacni_systemy/mikrosystemy/logo/zakladni_pristroje/manual_logo_0ba5_2005_cz.pdf